BigNurse: A Wireless Ad Hoc Network for Patient Monitoring

Roland Bader, Michele Pinto, Felix Spenrath, Philipp Wollmann, and Frank Kargl
Media Informatics Institute, Ulm University, Ulm, Germany

***Abstract*— The world faces a tremendous increase of elderly people with disabilities and a resulting shortages in the health care system which cannot keep pace with the ongoing increase of people over 65. In addition, stays in hospitals attached to beeping machines normally is a very uncomfortable time for patients.**

Ubiquitous Computing technology promises to change that. Our project gives people the opportunity to be mobile while their vital functions are monitored, thanks to localization they can be found and helped immediately on emergencies and doctors can observe patients wherever they are. Medical facilities would be relieved by such a system and the patients' quality of life would increase.

I. Introduction

The elderly population is increasing worldwide and it is expected that the number of people over 65 will increase even more in the coming decades. Many elderly people suffer from physical and mental disabilities that affect their everyday life. This often makes it necessary to constantly monitor the vital functions of these patients in hospitals or nursery homes which is both costly and uncomfortable.

For years, there have been considerations to improve patient monitoring both at home or at the hospital. *Ubiquitous Computing* now offers a way to solve some of the related problems. Using a *Ubiquitous Computing Patient Monitoring System*, patients are no longer forced to stay in hospitals, or at least not in bed in order to be in reach of the monitoring equipment.

We are currently designing and developing such a system with primary focus on use in hospitals or for clinical studies in a controlled home environment. Our system is called "'BigNurse" and is based on *MicaZ* motes [5], [1] that the patient will carry in its pocket. Motes are small computers used for building up *Wireless Sensor Networks*. They possess both sensing capabilities and connectivity using a wireless ZigBee [3] interface. Several sensors like oxygen saturation, pulse, blood pressure, brain waves, or muscle activity are attached to the mote which uses a multi-channel ADC sensor board to monitor many signals in parallel. These values are then transmitted to a base station where they are recorded, displayed and analyzed.

A physician or nurse is easily able to monitor all the patients in a medical station from one place. They can both access real-time data as well as a recorded history of any value. Thresholds will automatically trigger an alarm in case of critical situations. If patients roam about freely in the hospital and faint e.g. in the toilets, locating these patients in case of an alarm may become difficult if staff members need to search manually.

Therefore, "BigNurse" also implements *automatic localization* to find patients in emergencies. The system also provides *multi-hop networking*, so the network can extend over a much larger area than the direct ZigBee radio range and can cover whole floors or buildings.

The system was designed and developed in cooperation with professionals at the clinic for dentistry at Ulm University. They gave the input for the basic concept of the system and built the specific hardware to connect the sensor.

The next sections now describe some details regarding the different components of "BigNurse".

II. Architecture

A. General

"BigNurse" consists of two main components: the software that runs on the wireless sensor network nodes and the server application that collects, analyzes and displays data. Figure 1 shows the overall architecture of the mote-application. The wireless motes run the TinyOS operating system [2]. Applications consist of several modules that are written in a special C dialect called nesC. Applications are build by wiring these modules together via interfaces.

There are two kinds of motes in the system: *mobile motes* collect and pre-process sensor-readings and send this data to the server or relay motes. At the same time, the mobile motes should also act as multi-hop relays to enhance connectivity. Due to the restricted resources of the motes, this is not yet implemented in the prototype, but is planned for a future version.

The *beacon motes* support the localization mechanism. They are installed at fixed positions known to the server and emit beacons on request from the server. Mobile motes receive these beacons and send the received signal strength to the server where the mote position gets calculated. The beacon motes also act as multi-hop relays to enhance connectivity.

The server application is implemented in Java. Sensor data received from the network is collected in a database and can then be displayed for analysis. Additionally, it is possible to control the settings of motes over the PC by first requesting the mote settings and then sending the modified ones back. Alarm thresholds can be set globally or individually per patient and alarms are triggered accordingly. Localized motes are displayed on a map. Localization is either started automatically or in case of an alarm.

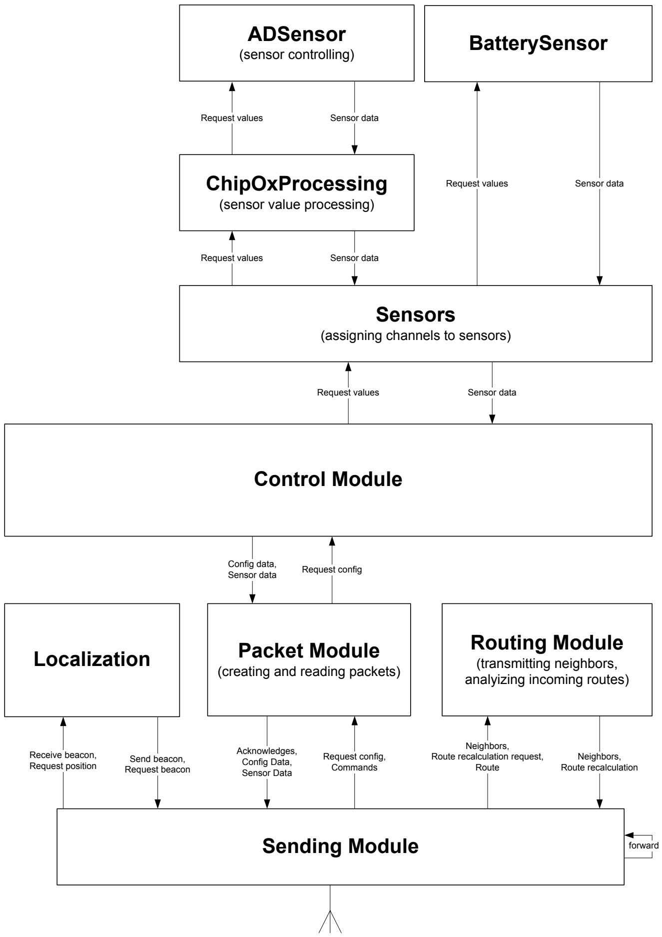

Fig. 1. Overall mote architecture

B. Sensors

Our architecture supports multiple sensors to be connected to the mote. Currently, we use a combined pulse-oximeter sensor for testing. This sensor is attached to the a MDA300 sensor board which offers 8 analog inputs channels. The sensor has a multiplexed analog output for all the measured data. As the multiplexing rate exceeds the capabilities of the MDA300, we connected the sensor via a custom sample and hold circuit that provides the sensor values on separate channels to the MDA300.

C. Multi-hopping

For multi-hop routing we initially tried to use the multi-hop components provided with the TinyOS ”Surge” application. In order to enhance performance and safe energy, we adapted this approach in several ways.

In contrast to the TinyOS approach, where motes maintain and calculate the routes, the route calculation in ”BigNurse” is done on the more powerful and less energy-constrained server. All motes start in broadcast mode, where sensor values are simply flooded in the network until they reach the gateway. During this flooding, nodes collect information on their neighbors which are afterwards sent to the gateway. The gateway collects these messages and calculates a spanning tree which is again flooded in the network. Every node receiving this routing message will simply set its next hop according to the information contained.

This saves energy and resources on the motes, while the server has full knowledge of the whole network and can calculate optimal routes. In a future version, the server should also consider additional factors like load or energy status of the motes. In case of topology changes, the mote detecting these changes switches back to broadcast mode and sends new topology information to the server which will then recalculate the routes.

Of course this approach assumes a rather static environment, where positions of motes change rather seldom. This may be a valid assumption for a clinical department, where patients tend to stay in their rooms and move only from time to time. In that case, some recalculations are acceptable.

D. Server application

The server application serves two purposes:

1) It is the interface between the wireless sensor network and the database. All patient mote data is saved immediately in the database when packets arrive from the mote network. This minimizes the risk that data gets lost e.g. because of server crashes. The reliability could have been enhanced further when data packets would have to be acknowledged before the data is deleted from the motes. As this will cause additional network overhead and energy consumption and as the data storage capacity of the motes is very restricted, we did not implement this option.
2) The application also contains a graphical user interface, where an observer can monitor the sensor values of the patients in real time and will be alarmed if an emergency occurs. The operator can then initiate additional actions, like increasing the sensing rate of that mote or requesting a localization of that mote which will be shown on a map.

III. Localization

As outlined in the motivation, locating patients quickly in case of emergencies may be of vital importance. We have integrated a localization mechanism that offers the following advantages:

- The localization is completely integrated into the wireless sensor network. No additional components (like RFIDs, ultrasonic devices, etc.) are needed.
- Localization is based on received signal strength measurements from beacon motes.
- Localization is done only on request (e.g. in case of emergencies). This prevents excessive collection of position traces which would affect patient's privacy.

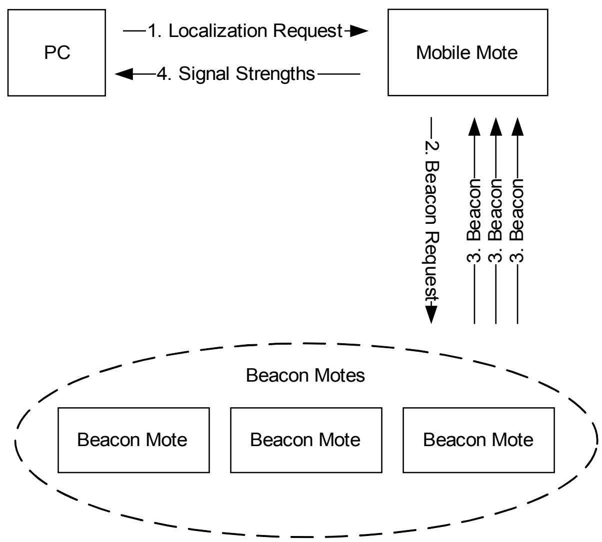

Fig. 2. basic principle of our localization

A. Basic operation

The localization is based on ”beacon motes”, which are evenly distributed in the building. They should be placed in a way that three beacon motes can be received in the whole monitored area. The position of these beacon motes is well-known and stored on the server. When the server requests a localization, it first sends an unicast localization request to the mobile mote to be localized. Next, this mote sends a broadcast localization request which is received by all nearby beacon motes. The beacon motes then emit beacons, which are again received by the mobile mote to be localized.

When receiving these beacons, the mobile mote measures the individual signal strengths. It sends these values to the server, which is then able to calculate the position of the mobile mote. So the localization consists of 4 steps which are visualized in Figure 2:

1) The PC sends a localization request to the mobile mote for which the position should be determined.
2) When receiving a localization request the mobile mote broadcasts a localization request to all beacon motes within range.
3) The beacon motes send several beacon messages which are received by the mobile mote.
4) The measurements of received signal strengths is sent to the server, where the position of the mobile mote is calculated and displayed.

Of course the broadcast localization requests are not forwarded by the multihop protocol, because they are irrelevant for all motes outside RF range. Unicast localization requests and beacon messages are transmitted as unicast.

B. Collecting localization data on the motes

After receiving the unicast localization request the mobile mote sends a broadcast localization request and starts a timer which fires after 2.8 seconds. The beacon motes receiving this request wait a short random delay and then send four beacon messages every 500ms. Thus the mobile mote collects from every beacon mote at most five beacon messages which are averaged and sent back to the server.

C. Calculating the motes’ position on the PC

As the ZigBee-range is very limited, we decided to not use a pre-recorded map of reference signatures like it is e.g. done in MoteTrack [4]. Creating these maps is a very time consuming task, they need to be updated, when e.g. furniture is re-arranged, and they are usually only valid for one dedicated receiver hardware. Because the system was primarily built for the application in a clinical environment, reference measurements are especially impractical. We think that the additional gain of accuracy is not worth the effort because you need a lot of measurements to really get a significant better accuracy. In contrast, our system is simple to deploy, by adding deployed beacon motes on the map, and provides sufficient accuracy to locate a patient as you can see in section III-E.

The position of the mobile motes to be localized is calculated by comparing the received signal strengths of the beacon motes. We make the assumption that for any two beacons b_i and b_j received from beacon motes i and j by the mobile mote m, the ratio between the squares of the two distances $d_m i$ and $d_m j$[1] is reciprocally proportional to the ratio between the signal strengths $rssi_i$ and $rssi_j$ received from these beacons:

$$\frac{d_i^2}{d_j^2} = n * \frac{rssi_j}{rssi_i}$$

We do this comparison for each pair of beacon motes from which we receive beacons and estimate the position on the map, where these ratios match best.

There is however a problem with this approach: if all beacon motes are placed in a small part of the map, the best estimated position can be in a very remote part of the map. So we have to make the assumption that the beacon motes are positioned equally on the map and that there are sufficient of these motes positioned on the outer edge of the area to be covered. Additionally, we prefer points in our algorithm that are close to the beacon motes.

In our approach walls and other obstacles are not taken into account for the calculation. This could be improved in a future version.

D. Security and privacy

As security wasn’t part of our project, all data is beeing sent unencrypted. However encryption can easily be added to our system. Basic privacy is given by the distribution of the beacon motes. Usually the position and the IDs of the beacon motes

[1] $d_m i$ denotes the distance from the mobile mote m to the beacon mote i

are only known by the server. The position is calculated on the server, so only the received signal strengths are sent over the network. However these informations are useless to someone who doesn't know the arrangement of the beacon motes.

E. Evaluation

In order to evaluate our approach, we have tested our implementation in a scenario where we distributed 5 beacon motes in a part of our lab. The beacons were placed every three to four meters in a hallway (approx. 20 meter length). Details can be seen in Figure 3. Tests in a larger area are planned, but need to be delayed until a larger number of motes will be available in our lab.

Besides the placement of the beacon motes, the figure also shows the difference between real and estimated positions of mobile motes. In this situation the error was always less than 4 meters. Thinking of the typical layout of clinical buildings, this should be sufficient to quickly find a patient in an emergency, as only one or two rooms must be searched.

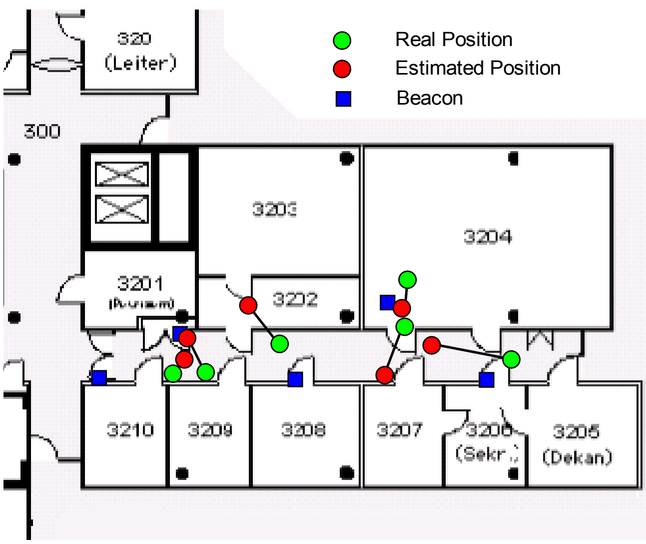

Fig. 3. Localization Evaluation

IV. Conclusion and Future Work

In this paper, we have presented the design of and first experiences with a health-monitoring wireless sensor network based on Micaz motes. Our prototype allows the monitoring and collection of arbitrary medical parameters on a large number of patients in parallel.

These values may be viewed and analyzed in real-time or in retrospect. Alarms can be set that will be triggered at given thresholds. In that case, the position of patients can be localized with a sufficient precision to quickly assist in cases of emergency.

Our prototype works fine with a small number of motes in its current state. To make it useful for hospitals with a large number of patients, we need additional testing and enhancements. E.g. our system should be capable of handling more gateways and different servers that can be accessed remotely from the PCs of all staff members. This can enhance geographical coverage and reliability. For a real-world application, the mote database should be integrated with the patient database, so all relevant patient information can be shown in one program. Another idea is to add an alarm button to the mobile motes, so patients can call for help themselves.

After finishing and initially testing the implementation, a larger lab test is planned in a realistic scenario at the clinic for dentistry at Ulm University. There, the system should be used for their research on the sleeping behavior of patients.

References

[1] Inc. Crossbow Technology. Motes, smart dust sensors, wireless sensor networks. http://www.xbow.com/Products/productsdetails.aspx?sid=3 [18.07.2006].
[2] U.C. Berkeley EECS Department. Tinyos. http://www.tinyos.net/ [18.07.2006].
[3] IEEE. IEEE 802.15.4 (ZigBee). http://grouper.ieee.org/groups/802/15/pub/TG4.html [18.07.2006].
[4] Konrad Lorincz and Matt Welsh. Motetrack: A robust, decentralized approach to rf-based location tracking. Proceedings of the International Workshop on Location and Context-Awareness (LoCA 2005), May 2005.
[5] Berkeley Wireless Embedded Systems, University of California. Network Embedded Systems Technology (NEST). http://webs.cs.berkeley.edu/nest-index.html [31.07.2006].

Appendix

The "BigNurse" prototype has been developed as a result of a students project conducted by Roland Bader, Michele Pinto, Felix Spenrath, and Philipp Wollmann. Roland, Michele, Felix, and Philipp study computer science at Ulm University with Ubiquitous Computing as a major subject. Frank works as senior scientist at the Media Informatics and supervises the project. His research interests include Ubiquitous Computing systems, ad hoc networks, and security. As explained in the paper, we plan to develop application towards a state where our partners in the University clinic can use it for their clinical studies and in the future perhaps also for normal diagnostic and therapy of regular patients.